**Library Management System**

**Linear Search:**

Sequentially checks each element in the list until the target value is found or the end of the list is reached.

- Time Complexity: O(n) in Worst-Case (Target is not in the list or is the last element).

- Use Case: Suitable for unsorted or small datasets.

**Binary Search**

Divides a sorted list into halves repeatedly, comparing the target value with the middle element to determine which half to continue searching.

- Time Complexity: O(log n) in Worst-Case (Target is not in the list, but still requires logarithmic steps.)

- Use Case: Suitable for large, sorted datasets.

**Time Complexity Comparison**

* **Linear Search**:
  + **Best-Case**: O(1)
  + **Average-Case**: O(n)
  + **Worst-Case**: O(n)
* **Binary Search**:
  + **Best-Case**: O(1)
  + **Average-Case**: O(logn)
  + **Worst-Case**:O(logn)

**When to Use Each Algorithm**

* **Linear Search**:
  + **Use**: When data is small or unsorted.
* **Binary Search**:
  + **Use**: When data is large and sorted.